UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/598,714	03/08/2007	**Anders Berkeman**	P18536-US2	9238

27045 7590 06/22/2010

ERICSSON INC.
6300 LEGACY DRIVE
M/S EVR 1-C-11
PLANO, TX 75024

EXAMINER	
AHMED, HAMDY S	

ART UNIT	PAPER NUMBER
2186	

NOTIFICATION DATE	DELIVERY MODE
06/22/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

kara.coffman@ericsson.com
jennifer.hardin@ericsson.com
melissa.rhea@ericsson.com

Office Action Summary	Application No.	Applicant(s)	
	10/598,714	BERKEMAN, ANDERS	
	Examiner	Art Unit	
	HAMDY S. AHMED	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on *01 March 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *35-84* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *35-84* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

Claims1-34 are canceled.

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 35-59, 63-69, 73-82, and 84 are rejected under 35 U.S.C. 102(b) as being anticipated by Agrawal et al. patent No: (US 6,314,534 B1).

As to claim 35 Agrawal discloses a **method for generating an address value for addressing a memory which is an interleaver or deinterleaver memory** (see abstract, lines 1-2), **comprising the steps of generating a plurality of address fragments**, (see abstract, lines 3-6, where more than one is generated), **and comparing only a fraction of the generated address fragments** (see column 2, lines 62-63, wherein only a portion of addresses generated are used) **with a maximum allowable value,** [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)] **wherein generating a plurality of address fragments,** (see abstract, lines 3-6, where more than one is generated), **further comprises the step of generating a first address fragment for a first address**

value, [generating two consecutive candidates bit reversed fragments (abstract)] **and a second address fragment, which is consecutive of the first address fragment, for a second address value;** [generating two consecutive candidates bit reversed fragments (abstract)] **and wherein the step of comparing only a fraction of the generated address fragments with a maximum allowable value,** (see column 2, lines 62-63, wherein only a portion of addresses generated are used)**, further comprises the step of comparing only every other address fragment of the plurality of address fragments with the maximum allowable value** (see column 3, lines 42-49, wherein the control unit receives the column and row for the current address and determines if the resulting address (the concatenation of the two fragments) is greater than N, which is the number of data bits in the code interleaver)**, whereby the step of comparing comprises comparing the first address fragment with the maximum allowable value** [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)].

As to claim 36 , Agrawal discloses a **method further comprising the steps of: discarding the compared address fragment if it exceeds the maximum allowable value** (see column 3, lines 53-57, wherein if the current address is greater than N, the control unit effectively "punctures" the current address, which is the equivalent of discarding it)**; and accepting the compared address fragment otherwise** (see column 3, lines 48-52, wherein the control unit causes the multiplexer to output the current row address from the current row fragment generator to use it to generate the address; the multiplexer also outputs final column fragment used to generate the address)**.**

As to claims 37 , Agrawal discloses **wherein the address fragments to be**

compared are permuted prior to the step of comparing (see column 3, lines 22-26 and 29-36, wherein the current row address fragment and the current column address fragment are applied to multiplexers before being compared).

As to claims 38 Agrawal discloses a **method further comprising the step of appending at least one most significant bit(s) to any address fragment or a permuted address fragment** (see column 4, lines 4-20, wherein bit reversed address fragments are used to ensure that at least one of two consecutively generated fragments will be in range for a interleaver code block of size N, where N is not an integer power of two.)

As to claims 39, 48, 56, and 65 Agrawal discloses **a method, wherein the compared address fragment is an odd address fragment to which a 1 is to be appended as a most significant bit** (see column 4, lines 38-48, wherein the control unit asserts a control signal when it receives the final row from the row fragment generator and the row value received is equal to the first row processed for each column).

As to claims 40, 49, 57 Agrawal discloses a **method wherein an even address fragment is generated in response to a step of discarding or accepting the compared address fragment** (see column 3, lines 19-41 and 48-55, wherein both current and next (Row+1 and col+1) row and column address fragments are generated and applied to multiplexers; the current address fragments are concatenated and compared to N, and depending on the result, the current address is either discarded or accepted, and the next address is discarded or accepted as well).

As to claim 41, Agrawal discloses a **method further comprising the steps of: generating at least the odd address fragment** (see column 3, lines 19-41, wherein the row generator generates a bit reversed row fragment for the current address and the

current column generator generates a current pseudo-random address fragment, which both correspond to the first address fragment) **to be compared** (see column 3, lines 19-41, wherein only the current bit reversed address fragment and the output of the current column generator are applied to the control unit, where together they are compared to the maximum value) **and a following even address fragment** (see column 3, line 19-41, wherein the Row+1 generator generates a next bit reversed row fragment for the next address, and the Next column generator (col+1) generates a next pseudo-random address fragment for the next address) **during a first clock cycle**(see column 3, lines 57-61, wherein the method ensures that, if the current address is out of range, the next address will be in range and that a useable address will be available during the same clock cycle)**; if the compared odd address fragment is discarded, outputting the even address fragment during the first clock cycle** (see column 3, lines 53-57, wherein if the current address is greater than N, the control unit causes the multiplexer to output the row fragment from the Row+1 generator)**; if the compared odd address fragment is accepted, outputting the odd compared address fragment** (see column 3, lines 48-52, wherein the control unit causes the multiplexer to output the current row address from the current row fragment generator) **and retaining values of registers of a shift register during the first clock cycle** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle)**; and outputting the even address fragment during a second clock cycle following the first clock cycle** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle)**.**

As to claims 42 and 45, Agrawal discloses a **method further comprising the step of generating a next odd address fragment** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle)**; and inputting the next odd address fragment into registers of a shift register** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle)**.**

As to claims 43, 44, Agrawal discloses a **method wherein the next even and next odd address fragments are generated by means of a feedback function (g(x))** (see column 6, lines 21- 33, wherein look up tables receive current row and column indices from the address generators and generate values, which in turn are used in calculations to produce values that are submitted to multiplexers that output final row and column indices)**.**

As to claims 46 Agrawal discloses a **method further comprising the steps of: discarding the compared address fragment if it exceeds the maximum allowable value** (see column 3, lines 53-57, wherein if the current address is greater than N, the control unit effectively "punctures" the current address, which is the equivalent of discarding it)**; accepting the compared address fragment otherwise** (see column 3, lines 48-52, wherein the control unit causes the multiplexer to output the current row address from the current row fragment generator to use it to generate the address; the multiplexer also outputs final column fragment used to generate the address); **and permuting the generated address fragments** (see column 5, lines 15-25)**, wherein the address fragments to be compared are permuted prior to the step of**

comparing (see column 3, lines 22-26 and 29-36, wherein the current row address fragment and the current column address fragment are applied to multiplexers before being compared).

As to claim 47 Agrawal discloses a **method further comprising the step of appending at least one most significant bit(s) to any address fragment or a permuted address fragment** (see column 4, lines 4-20, wherein bit reversed address fragments are used to ensure that at least one of two consecutively generated fragments will be in range for a interleaver code block of size N, where N is not an integer power of two.)

As to claims 50 Agrawal discloses **a method for generating an address value for addressing a memory which is an interleaver or deinterleaver memory** (see abstract, lines 1-2) (see abstract, lines 3-6, where more than one is generated), (see column 2, lines 62-63, wherein only a portion of addresses generated are used), **comprising the steps of: generating a plurality of address fragments,** (see abstract, lines 3-6, where more than one is generated) **by generating a first address fragment for a first address value,** (see column 3, lines 19-41, wherein the row generator generates a bit reversed row fragment for the current address and the current column generator generates a current pseudo-random address fragment, which both correspond to the first address fragment)**, and a second address fragment, which is consecutive of the first address fragment, for a second address value** (see column 3, line 19-41, wherein the Row+1 generator generates a next bit reversed row fragment for the next address, and the Next column generator (col+1) generates a next pseudo-random address fragment for the next address)**; and comparing only a fraction of the generated address fragments,** (see column 2, lines 62-63, wherein only a portion of

addresses generated are used) **comparing only every other address fragment of the plurality of address fragments** (see column 3, lines 42-43, wherein the control unit receives only the column and row for the current address, and uses only those for comparison) **with stored address fragments, which are known to be out of range when permuted and comparing the first address of fragment with the stored address fragments** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle).

As to claims 51 Agrawal discloses **a method further comprising the step of permuting the generated address fragments** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle) **after the step of comparing.**

As to claims 52,, Agrawal discloses a **method for generating an address value for addressing a memory which is an interleaver or deinterleaver memory** (see abstract, lines 1-2), **comprising a means for** a means for generating a plurality of address fragments configured to generate a first address fragment for a first address value and a second address fragment [generating two consecutive candidates bit reversed fragments (abstract)]:, which is consecutive of the first address fragment, for a second address value; [generating two consecutive candidates bit reversed fragments (abstract)] and a comparator means configured to compare only a fraction of the plurality of address fragments with a maximum allowable value and further configured to compare only every other address fragment of the plurality of address

fragments with the maximum allowable value and thereby to compare the first address fragment with the maximum allowable value [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)].

As to claim 53, Agrawal discloses a **device comprising a selector means configured to discard the compared address fragment if it exceeds the maximum allowable value** (see column 3, lines 53-57, wherein if the current address is greater than N, the control unit effectively "punctures" the current address, which is the equivalent of discarding it)**; and to accept the compared address fragment otherwise.** (see column 3, lines 48-52, wherein the control unit causes the multiplexer to output the current row address from the current row fragment generator to use it to generate the address; the multiplexer also outputs final column fragment used to generate the address)**.**

As to claim 54, Agrawal discloses a **method further comprising a permuting means configured to permute address fragments** (see column 5, lines 15-25)**, the permuting means being provided prior to the comparator means.** (see column 3, lines 22-26 and 29-36, wherein the current row address fragment and the current column address fragment are applied to multiplexers before being compared)**.**

As to claim 55, Agrawal discloses a **method further comprising a toggle means configured to append at least one most significant bit(s) to any address fragment, or to any permuted address fragment, in order to generate the address value.** (see column 4, lines 4-20, wherein bit reversed address fragments are used to ensure that at least one of two consecutively generated fragments will be in range for a interleaver code block of size N, where N is not an integer power of two.)

As to claims 56 Agrawal discloses **a method, wherein the compared address fragment is an odd address fragment to which a 1 is to be appended as a most significant bit** (see column 4, lines 38-48, wherein the control unit asserts a control signal when it receives the final row from the row fragment generator and the row value received is equal to the first row processed for each column)**.**

As to claim 57 Agrawal discloses a **method wherein the means for generating address fragments is configured to generate a next even address fragment in response to a discarding or acceptance of the compared address fragment.** (see column 3, lines 19-41 and 48-55, wherein both current and next (Row+1 and col+1) row and column address fragments are generated and applied to multiplexers; the current address fragments are concatenated and compared to N, and depending on the result, the current address is either discarded or accepted, and the next address is discarded or accepted as well)**.**

As to claims 58 and 67, Agrawal discloses a **device wherein the means for generating address fragments**(see abstract, lines 3-6, where more than one is generated) **further comprises: a shift-register comprising a predetermined number of registers configured to generate address fragments to be compared during a first clock cycle** (see figure 3, element 400, and column 3, lines 43-63, wherein within the bit reversed address fragment generator, a row counter cycles from 0 to r-1, for the set of r rows in the interleaver memory)**; an address fragment calculation means configured to generate a next even address fragment during the first clock cycle** (see column 3, lines 19-41, wherein the next row (row+1) and next column (col+1) generators generate a next bit reversed row fragment and next pseudo-random address fragment, respectively, for the next address)**, which is based on the address fragment to be compared** (see column 3, lines 62-67, and column 4, lines 1-12,

wherein using rows of bit reversed addresses means that the most significant bit of the two address fragments alternates between logic zero and logic one for each new address; since the least significant bit becomes the most significant bit during reversal, the most significant bit in the bit reversed address increments each time. By generating bit reversed address fragments for the current address and the next address, it is ensured that at least one address will be in range and available during that clock cycle); **a selector means is configured to, if the compared address fragment is discarded, output the even address fragment in response to a first control signal (M) during the first clock cycle** (see column 3, lines 53-55, wherein if the current address is greater than N, the control unit causes the multiplexer to output the row fragment from the row+1 generator), **and to output the compared address fragment during the first clock cycle if the compared address fragment is accepted** (see column 3, lines 42-52, wherein if the current address is not greater than the size of the code interleaver block, the control unit causes the multiplexer to output the current row address from the current row fragment generator), **and to output the even address fragment during a second clock cycle following the first clock cycle** (see column 6, lines 4-7, wherein the multiplexer outputs the next row index value (the unreversed next row value0 which is used to generate the current row and next row during the next cycle); **and the shift register is configured to retain present values of the registers during the first clock cycle in response to a second control signal (E) if the compared address fragment is accepted** (see column 7, lines 4-11, wherein if a row index is punctured, but the column index remains the same, a signal causes multiplexers to output the next row value of the associated set of input values, and other multiplexers cause the I) value to be used when j=c-1 to implement the processing steps described above to produce the final row index and the final column index, which are concatenated to

generate the address used for interleaving or deinterleaving).

As to claims 59 , Agrawal discloses a **device wherein the device is implemented by software comprising readable program means to be run by a processor** (inherently, a device and method that generates address fragments and compares them to a maximum value is comprised of a processor that is using a software program to perform such functions).

As to claim 63, Agrawal discloses a **method further comprising a selector means configured to discard the compared address fragment if it exceeds the maximum allowable value,** (see column 3, lines 53-57, wherein if the current address is greater than N, the control unit effectively "punctures" the current address, which is the equivalent of discarding it)**; and to accept the compared address fragment otherwise;** (see column 3, lines 48-52, wherein the control unit causes the multiplexer to output the current row address from the current row fragment generator to use it to generate the address; the multiplexer also outputs final column fragment used to generate the address); **and a permuting means configured to permute the address fragments, the permuting means being provided prior to the comparator means.** (see column 3, lines 22-26 and 29-36, wherein the current row address fragment and the current column address fragment are applied to multiplexers before being compared).

As to claim 64, Agrawal discloses a **method further comprising toggle means configured to append at least one most significant bit(s) to any address fragment or a permuted address fragment , in order to generate the address value.**(see column 4, lines 4-20, wherein bit reversed address fragments are used to ensure that at least one of two consecutively generated fragments will be in range for a interleaver code block of size N, where N is not an integer power of two.)

As to claim 65, Agrawal discloses **a method, wherein the compared address fragment is an odd address fragment to which a 1 is to be appended as a most significant bit** (see column 4, lines 38-48, wherein the control unit asserts a control signal when it receives the final row from the row fragment generator and the row value received is equal to the first row processed for each column)**.**

As to claim 66, Agrawal discloses a **method wherein the means for generating address fragments is configured to generate a next even address fragment in response to discarding or accepting the compared address fragment.** (see column 3, lines 19-41 and 48-55, wherein both current and next (Row+1 and col+1) row and column address fragments are generated and applied to multiplexers; the current address fragments are concatenated and compared to N, and depending on the result, the current address is either discarded or accepted, and the next address is discarded or accepted as well)**.**

As to claim 67, Agrawal discloses a **device wherein the means for generating address fragments** (see abstract, lines 3-6, where more than one is generated) **further comprises: a shift-register comprising a predetermined number of registers configured to generate address fragments to be compared during a first clock cycle;** (see figure 3, element 400, and column 3, lines 43-63, wherein within the bit reversed address fragment generator, a row counter cycles from 0 to r-1, for the set of r rows in the interleaver memory)**; address fragment calculation means configured to generate a next even address fragment during the first clock cycle,** (see column 3, lines 19-41, wherein the next row (row+1) and next column (col+1) generators generate a next bit reversed row fragment and next pseudo-random address fragment, respectively, for the next address)**, which is based on the address fragment to be compared;** (see column 3, lines 62-67, and column 4, lines 1-12, wherein using rows of

bit reversed addresses means that the most significant bit of the two address fragments alternates between logic zero and logic one for each new address; since the least significant bit becomes the most significant bit during reversal, the most significant bit in the bit reversed address increments each time. By generating bit reversed address fragments for the current address and the next address, it is ensured that at least one address will be in range and available during that clock cycle)**; the selector means is configured to, if the compared address fragment is discarded, output the even address fragment in response to a first control signal (M) during the first clock cycle,** (see column 3, lines 53-55, wherein if the current address is greater than N, the control unit causes the multiplexer to output the row fragment from the row+1 generator)**, and to output the compared address fragment during the first clock cycle if the compared address fragment is accepted,** (see column 3, lines 42-52, wherein if the current address is not greater than the size of the code interleaver block, the control unit causes the multiplexer to output the current row address from the current row fragment generator)**, and to output the even address fragment during a second clock cycle following the first clock cycle** (see column 6, lines 4-7, wherein the multiplexer outputs the next row index value (the unreversed next row value0 which is used to generate the current row and next row during the next cycle)**; and the shift register is configured to retain present values of the registers during the first clock cycle in response to a second control signal (E) if the compared address fragment is accepted** (see column 7, lines 4-11, wherein if a row index is punctured, but the column index remains the same, a signal causes multiplexers to output the next row value of the associated set of input values, and other multiplexers cause the I) value to be used when j=c-1 to implement the processing steps described above to

produce the final row index and the final column index, which are concatenated to generate the address used for interleaving or deinterleaving).

As to claim 68, Agrawal discloses a **device wherein the address fragment calculation unit mean is configured to generate a next odd address fragment, based on the even address fragment,** (see column 6, lines 4-7, wherein the next row index value (the even address fragment) is used to generate the current row (odd) and next (even) row during the next cycle) **and feed back said next odd address fragment to the shift register.** (see column 6, lines 4-7, the values generated are kept to be used to generate the subsequent addresses in the following clock cycle).

As to claim 69 Agrawal discloses a **method wherein address fragment calculation means is configured to generate the next even and next odd address fragments by means of a feedback function.** (see column 6, lines 21- 33, wherein look up tables receive current row and column indices from the address generators and generate values, which in turn are used in calculations to produce values that are submitted to multiplexers that output final row and column indices).

As to claim 73, Agrawal discloses a **method wherein the next address fragment calculation means is configured to generate the next even and next odd address fragments by means of a feedback function.** (see column 6, lines 21- 33, wherein look up tables receive current row and column indices from the address generators and generate values, which in turn are used in calculations to produce values that are submitted to multiplexers that output final row and column indices).

As to claim 74, Agrawal discloses a **device wherein the device is implemented by software comprising readable program loaded in a computer readable medium and executed by a processor.** (inherently, a device and method

that generates address fragments and compares them to a maximum value is comprised of a processor that is using a software program to perform such functions).

As to claim 75, Agrawal discloses **a method for generating an address value for addressing a memory which is an interleaver or deinterleaver memory** (see abstract, lines 1-2) **comprising: a means for generating a plurality of address fragments configured to generate a first address fragment for a first address value and a second address fragment, which is consecutive of the first address fragment, for a second address value; and** (see abstract, lines 3-6, where more than one is generated), **a comparator means configured to compare only a fraction of the plurality of address fragments** (see column 2, lines 62-63, wherein only a portion of addresses generated are used)**with a maximum allowable value, further being configured to compare only every other address fragment of the plurality of address fragments with stored address fragments, which are known to be out of range when permuted, and thereby to compare the first address fragment with stored address fragment.**(see abstract lines 3-6, where more than one is generated), (see column 3, lines 19-41, wherein the row generator generates a bit reversed row fragment for the current address and the current column generator generates a current pseudo-random address fragment, which both correspond to the first address fragment), (see column 3, line 19-41, wherein the Row+1 generator generates a next bit reversed row fragment for the next address, and the Next column generator (col+1) generates a next pseudo-random address fragment for the next address); (see column 2, lines 62-63, wherein only a portion of addresses generated are used), (see column 3, lines 42-43, wherein the control unit receives only the column and row for the current address, and uses only those for comparison) (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index

value (the unreversed next row value) which is used to generate the current row and next row during the next cycle).

As to claim 76, Agrawal discloses **a method further comprising a permuting means configured to permute the address fragments, the permuting means being provided after the comparator means.** (see column 6, lines 4-7, wherein if the current row address fragment is accepted, the multiplexer outputs the next row index value (the unreversed next row value) which is used to generate the current row and next row during the next cycle).

As to claim, 77 Agrawal discloses a **method for interleaving a block of data, having a memory which is an interleaver memory in combination(**see abstract, lines 1-2), **and a device for generating address values, comprising: a means for generating a plurality of address fragments** (see abstract, lines 3-6, where more than one is generated), **and comparator means configured to compare only a fraction of the plurality of address fragments** (see column 2, lines 62-63, wherein only a portion of addresses generated are used) **with a maximum allowable value** (see column 3, lines 48-49, wherein the control unit determines if the current address is greater than the size of the code interleaver block, which is the equivalent of comparing the fragment size to a maximum allowable value), **the means for generating a plurality of address fragments (**see abstract, lines 3-6, where more than one is generated), **being configured to generate a first address fragment for a first address value** (see column 3, lines 19-41, wherein the row generator generates a bit reversed row fragment for the current address and the current column generator generates a current pseudo-random address fragment, which both correspond to the first address fragment), **and a second address fragment, which is consecutive of the first address fragment, for a second address value** (see column 3, line 19-41,

wherein the Row+1 generator generates a next bit reversed row fragment for the next address, and the Next column generator (col+1) generates a next pseudo-random address fragment for the next address); **and the comparator means being configured to** (see column 2, lines 62-63, wherein only a portion of addresses generated are used), **compare only every other address fragment of the plurality of address fragments with the maximum allowable value** (see column 3, lines 42-49, wherein the control unit receives the column and row for the current address and determines if the resulting address (the concatenation of the two fragments) is greater than N, which is the number of data bits in the code interleaver), **and thereby to compare the first address fragment with the maximum allowable value.**(see column 3, lines 19-41, wherein only the current bit reversed address fragment and the output of the current column generator are applied to the control unit, where together they are compared to the maximum value).

As to claim, 78 Agrawal discloses a **method for generating an address value for addressing a memory which is an interleaver or deinterleaver memory** (see abstract, lines 1-2), **comprising**: a means for **generating a plurality of address fragments** (see abstract, lines 3-6, where more than one is generated), **and comparator means configured to compare only a fraction of the plurality of address fragments** (see column 2, lines 62-63, wherein only a portion of addresses generated are used) **with a maximum allowable value** (see column 3, lines 48-49, wherein the control unit determines if the current address is greater than the size of the code interleaver block, which is the equivalent of comparing the fragment size to a maximum allowable value), **the means for generating a plurality of address fragments** (see abstract, lines 3-6, where more than one is generated), being configured to generate a first address fragment for a first address value (see column 3,

lines 19-41, wherein the row generator generates a bit reversed row fragment for the current address and the current column generator generates a current pseudo-random address fragment, which both correspond to the first address fragment), **and a second address fragment, which is consecutive of the first address fragment, for a second address value** (see column 3, line 19-41, wherein the Row+1 generator generates a next bit reversed row fragment for the next address, and the Next column generator (col+1) generates a next pseudo-random address fragment for the next address); **and the comparator means being configured to compare only every other address fragment of the plurality of address fragments with the maximum allowable value** (see column 2, lines 62-63, wherein only a portion of addresses generated are used)**, and thereby to compare the first address fragment with the maximum allowable value.** (see column 3, lines 42-49, wherein the control unit receives the column and row for the current address and determines if the resulting address (the concatenation of the two fragments) is greater than N, which is the number of data bits in the code interleaver)**,** (see column 3, lines 19-41, wherein only the current bit reversed address fragment and the output of the current column generator are applied to the control unit, where together they are compared to the maximum value)**.**

As to claim 79 Agrawal discloses a communication apparatus for communicating data, comprising a memory which is an interleaver memory or a deinterleaver memory see abstract, lines 1-2), and a device for generating address values for addressing the memory see abstract, lines 3-6, where more than one is generated, comprising: a means for generating a plurality of address fragments configured to generate a first address fragment for a first address value and a second address fragment (see column 3, lines 48-49, wherein the control unit determines if the current address is greater than

the size of the code interleaver block, which is the equivalent of comparing the fragment size to a maximum allowable value), which is consecutive of the first address fragment, for a second address value; and a comparator means configured to compare only a fraction of the plurality of address fragments with a maximum allowable value adapted configured to compare only every other address fragment of the plurality of address fragments with the maximum allowable value and thereby compare the first address fragment with the maximum allowable value (the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)]..

As to claim 80, Agrawal discloses a **communication apparatus wherein the communication apparatus is one selected from the group consisting of a mobile radio terminal, a pager, a communicator, an electronic organizer, and a smartphone.** (see column 1, lines 51-58, wherein the invention may be used in various types of digital communications including satellite based digital communications systems and terrestrial based digital communication systems; both such systems are wireless and use radio frequency electromagnetic signals to transmit other signals or data, which is true for the devices named in the claim).

As to claim 81, Agrawal discloses a **communication apparatus wherein the communication apparatus comprises a mobile telephone** (see column 1, lines 51-

56, wherein the invention may be used in various types of digital communications systems such as cellular telephone systems).

As to claims 82 Agrawal discloses a **method for generating an address value for addressing a memory which is an interleaver or deinterleaver memory** (see abstract, lines 1-2), **having the steps of generating a plurality of address fragments** (see abstract, lines 3-6, where more than one is generated), **and comparing only a fraction of the generated address fragments** (see column 2, lines 62-63, wherein only a portion of addresses generated are used) **with a maximum allowable value** [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)];

comprising the further steps of: when generating a plurality of address fragments (see abstract, lines 3-6, where more than one is generated), **generating a first address fragment for a first address value** [generating twoconsecutive candidates bit reversed fragments (abstract)]:

, **and a second address fragment, which is consecutive of the first address fragment, for a second address value** [generating two consecutive candidates bit reversed fragments (abstract)] **and when comparing only a fraction of the generated address fragments with a maximum allowable value** (see column 2, lines 62-63, wherein only a portion of addresses generated are used)**, comparing only every other address fragment of the plurality of address fragments with the maximum allowable value** (see column 3, lines 42-49, wherein the control unit receives the column and row for the current address and determines if the resulting address (the concatenation of the two fragments) is greater than N, which is the number of data bits

in the code interleaver)**, whereby the step of comparing comprises comparing the first address fragment with the maximum allowable value** [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)].

As to claim 84, Agrawal discloses A software program product embodied on a computer readable medium executable by computer hardware when the product is executed by a processor contained in said computer hardware (see abstract), the product having instructions comprising **generating a plurality of address fragments** (see abstract, lines 3-6, where more than one is generated), **and comparing only a fraction of the generated address fragments** (see column 2, lines 62-63, wherein only a portion of addresses generated are used) **with a maximum allowable value** [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)];**comprising the further steps of: when generating a plurality of address fragments** (see abstract, lines 3-6, where more than one is generated), **generating a first address fragment for a first address value** [generating two consecutive candidates bit reversed fragments (abstract)], **and a second address fragment, which is consecutive of the first address fragment, for a second address value** [generating two consecutive candidates bit reversed fragments (abstract)] **and when comparing only a fraction of the generated address fragments with a maximum allowable value** (see column 2, lines 62-63, wherein only a portion of addresses generated are used)**, comparing only every other address fragment of the plurality of address fragments with the maximum allowable value** (see column 3, lines 42-49, wherein the control unit receives the column and row

for the current address and determines if the resulting address (the concatenation of the two fragments) is greater than N, which is the number of data bits in the code interleaver)**, whereby the step of comparing comprises comparing the first address fragment with the maximum allowable value** [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)].

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 60, 61, 70, 71 and 83 are rejected under 35 U.S.C. 103(a) as being unpatentable over Argawal et al Patent No: (US 6,314,534 B1), in view of Chang patent No: 5,687,325).

As to claims 60 and 70, Argawal teaches claim 50 as mention above. But Argawal does not teach **a device wherein the device is implemented as an application specific integrated circuit.** However Chang discloses a**dvice wherein the device is implemented as an application specific integrated circuit** (see abstract, lines 1-3, wherein an application-specific field programmable gate array includes at least two fixed functional units in a single IC chip)**.** Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the

method of Argawal by adopting the teaching of Chang by having a **device wherein the device is implemented as an application specific integrated circuit** in order for the fixed function together with PPGA perform all the functions specified for a particular ASIC design.

As to claims 61 and 71, Chang discloses a **device wherein the device is implemented as a field programmable gate array** (see abstract, lines 1-3, wherein an application-specific field programmable gate array includes at least two fixed functional units in a single IC chip)**.**

As to claim 83, Argawal teaches claim 82 as mention above. But Argawal does not disclose **a communication apparatus wherein the communication apparatus is one selected from the group consisting of a set-top-box, a television set, and a mobile television receiver.** However Chang discloses **a communication apparatus wherein the communication apparatus is one selected from the group consisting of a set-top-box, a television set, and a mobile television receiver** (see column 1, lines 20- 24, wherein an application-specific integrated circuit interfaces between a digital computer system's bus and a peripheral device such as a Video-on-Demand set-top box)**.** Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the method of Argawal by adopting the teaching of Chang by having **a communication apparatus wherein the communication apparatus is one selected from the group consisting of a set-top-box, a television set, and a mobile television receiver,** in order to reduce the cost of implementing a large number of digital logic circuit to perform a particular function.

Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Argawal et al Patent No: (US 6,314,534 B1), in view of Kurihara patent No: 5,128,998).

As to claim 62 Argawal teaches claim 50 as mention above. Argawal does not disclose a device further comprising a shift register configured to generate a maximum length pseudo noise sequence. However Kurihara discloses a device further comprising a shift register configured to generate a maximum length pseudo noise sequence (see column 2, lines 40-47, wherein when a maximum length linearly recurring sequence code is used for the pseudo noise code, a maximum length linearly recurring sequence code generator is used to generate the pseudo noise code; this type of generator contains flip-flops that comprise a shift register). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the method of Argawal by adopting the teaching of Kurihara by having a device further comprising a shift register adapted to generate a maximum length pseudo noise sequence, because the shift register is necessary as initial information of generating the code.

Claim 72 is rejected under 35 U.S.C. 103(a) as being unpatentable over Argawal et al Patent No: (US 6,314,534 B1), and Chang patent No: 5,687,325), in view of Kurihara patent No: 5,128,998).

As to claim 72, Argawal and Chang teach claim 71 as mention above. Argawal and Chang do not disclose a device further comprising a shift register configured to generate a maximum length pseudo noise sequence. However Kurihara discloses a device further comprising a shift register configured to generate a maximum length pseudo noise sequence (see column 2, lines 40-47, wherein when a maximum length linearly recurring sequence code is used for the pseudo noise code, a maximum length linearly recurring sequence code generator is used to generate the pseudo noise code; this type of generator contains flip-flops that comprise a shift register). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to

modify the method of Argawal and Chang by adopting the teaching of Kurihara and having a device further comprising a shift register adapted to generate a maximum length pseudo noise sequence, because the shift register is necessary as initial information of generating the code.

Response to Arguments

The applicant argues that Agrawal does not discloses the step of comparing only every other address fragment of the plurality of address fragments with the maximum allowable value, whereby the step of comparing comprises comparing the first address fragment with the maximum allowable value. However Agrawal shows [the first bit reversed address fragment generates an address that is greater than a maximum address (abstract); that is, only the first fragment is compared to the maximum address (abstract)].

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HAMDY S. AHMED whose telephone number is (571)270-1027. The examiner can normally be reached on M-TR 7:30-5:00pm and Every 2nd Friday 7:30-4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Matt Kim/
Supervisory Patent Examiner, Art Unit 2186

/Hamdy S Ahmed/
Examiner, Art Unit 2186